Enhancing Interactive Web Applications in Hybrid Networks

Aruna Balasubramanian Brian Neil Levine Arun Venkataramani
Department of Computer Science
University of Massachusetts Amherst
{arunab, brian, arun}@cs.umass.edu

ABSTRACT

Mobile Internet users have several options today including high bandwidth cellular data services such as 3G, that may be the choice for many. However, the ubiquity and low cost of WiFi suggests an attractive alternative, namely, opportunistic use of open WiFi access points (APs) or planned municipal mesh networks. Unfortunately, for vehicular users, the intermittent nature of WiFi connectivity makes it challenging to support popular interactive applications such as Web search and browsing. Our work is driven by two questions. 1) How can we enable system support for interactive web applications to tolerate disruptions in WiFi connectivity from mobile nodes? 2) Can opportunistic mobile-to-mobile (m2m) transfers enhance application performance over only using APs, and if so, under what conditions and by how much?

We present Thedu, a system that enables access to Web search from moving vehicles. The key idea is to use aggressive prefetching to transform the interactive Web search application into a one-shot request/response process. We deployed a prototype of Thedu on the DieselNet testbed in Amherst, MA, consisting of transit buses averaging 21 on the road at a time. Our deployment results show that Thedu can deliver 4 times as many relevant web pages than not using Thedu. A bus receives relevant web pages with a mean delay of 2.3 minutes and within 0.55 minutes in areas with high AP density. Thedu augments AP connectivity with m2m transfers using a utility-driven DTN routing algorithm and uses caching to exploit query locality. Our analytic model and trace-driven simulations suggest that m2m routing yields little benefit over using APs alone even under moderately dense AP deployment such as in Amherst. With sparsely deployed APs as may be the case in rural areas, our conclusions are more mixed: m2m routing with caching improves the number of relevant responses delivered per bus by up to 58%, but the mean delay is significantly high at 6.7 minutes, calling into question its practicality for interactive applications.

This work was supported in part by NSF awards CNS-0133055, CNS-0519881, and CNS-0721779 and in part by ARO award W911NF-07-1-0281.

MobiCom'08, September 14–19, 2008, San Francisco, California, USA.

Categories and Subject Descriptors

C.2 [**Computer Communication Network**]: Network Architecture and Design—*Wireless communication, Store and forward networks*

General Terms

Design, Performance

Keywords

Mobile networking, web search, application, testbed

1. INTRODUCTION

The value of mobile, networked systems is derived from a balance of the breadth of applications that can be supported and the costs to the user. Commercial cellular data services such as 3G provide good performance for a wide range of applications, but they incur monthly subscription fees and are limited to areas where the market supports the deployment of a cell tower infrastructure. For many, cellular services are the best choice. However, the ubiquity and low cost of WiFi suggests an attractive alternative, namely, opportunistic use of open WiFi Internet access points (APs) or planned municipal mesh networks. Unfortunately, for vehicular users, the intermittent nature of WiFi connectivity is inadequate to support popular interactive applications such as Web search and browsing. Our position is that the performance of such interactive applications can be significantly enhanced by providing system support to tolerate intermittent disconnections.

In this paper, we experimentally demonstrate the feasibility and performance of interactive Web search and retrieval using open-WiFi access in a mobile setting. Specifically, we address two questions. First, can intermittent *mobile-to-Internet AP* (m2i) connectivity be used to support interactive Web search and browsing? Second, can the performance or robustness of interactive applications be further enhanced by simultaneously leveraging opportunistic *mobile-to-mobile* (m2m) contacts?

Our experiments are based on a mobile system we have designed and deployed called *Thedu*. Currently, when Internet access is intermittent, Web search is cumbersome: a user issues a query, and a search engine returns a ranked list of URLs. Subsequently, the user clicks on one or more URLs and the search engine fetches the corresponding web page. Adapting this interactive process is not viable when periods of connectivity are few and disconnections can occur unexpectedly. Thedu addresses this problem using a proxy to aggressively prefetch web pages and transform the interactive search process into a transactional format. The mobile then downloads the prefetched web pages from the proxy over a series of contact opportunities. The challenge is in correctly prioritizing prefetched web pages during bandwidth-limited connections. During m2i connections, Thedu uses *(i)* a simple classifier and identifies queries that

likely require only a single web response and *(ii)* a novel scheme to normalize ranking of web pages across different search queries. Our experiments demonstrate that with application domain knowledge, simple changes can greatly improve performance for intermittently connected mobile networks.

To leverage m2m contacts, Thedu uses a utility-driven DTN routing algorithm that prioritizes and routes the most useful web pages to the mobile. During m2m routing, Thedu nodes also exploit query locality and cache popular web pages. Compared to m2i bandwidth, m2m bandwidth is typically limited. However, in some scenarios such as kiosk-based rural networks, the number of APs are limited and mobile nodes are plentiful in comparison. If the mobile nodes are willing to participate in routing—using a FON-like [13] incentive model or if they belong to a single organization such as a bus company—m2m contacts offer a significant opportunity to improve performance.

We measure the performance of Thedu using a real deployment and trace-driven simulation study on UMass DieselNet [3]. Our vehicular testbed consists of 40 buses in Amherst, MA, with about 21 on the road at a time. The buses transfer data among each other and with open APs found on the road. In our experiments, the average latency in receiving the relevant web pages for a query is dependent on where the mobile is located when the query is issued. In our experiments, for areas with a high density of open APs, queries are answered in 0.55 minutes; the overall mean is 2.3 minutes. Moreover, compared to not using Thedu, users can expect a fourfold increase in the number of useful web pages retrieved in response to queries. Note that our results measure the time from querying to completed retrieval of a *relevant* web page, rather than simply the search engine results page or retrieval of the first, possibly irrelevant, result listed.

Our trace-driven simulations show that in our testbed, where APs are densely deployed, m2m routing does not provide significant benefit compared to using m2i contacts alone. However, when buses are limited to using only 5 APs in our testbed to emulate a rural scenario, m2m routing with caching provides up to 58% increase in the number of relevant web pages delivered per bus. Unfortunately, the delay in receiving responses is significantly higher with a mean of 6.7 minutes, calling into question the practical applicability of m2m routing for interactive applications.

Our analytical model confirms these observations and suggests that m2m routing yields benefit only when APs are sparsely deployed. The benefit largely results from 1) the presence of fringe mobiles that do not come into frequent contact with APs, and 2) the decreasing marginal utility of fetching more documents,that makes it worthwhile for core mobiles to fetch documents for fringe mobiles, provided the delay is acceptable. Taken together, our experimental and analytical results limit the applicability of m2m routing only to delay-tolerant applications even in rural areas with limited infrastructure.

2. Thedu DESIGN ISSUES

An array of options are available to mobile users for connecting to the Internet today. Among other options, users can purchase high-speed 3G or GPRS service, or they can leverage open WiFi APs. The former generally offers better performance; however, there is a case for studying opportunistic mobile WiFi access.

WiFi hardware is commonly available, reasonably long-range, and is widely deployed in many non-mobile contexts (e.g., buildings, homes, and cafes). WiFi is currently an excellent radio platform to leverage for deploying a community-wide network, be it an ad hoc [16], community, or municipal effort. A significant number of users will pay a one-time cost for commodity WiFi equipment if (and only if) its performance is adequate overall. A primary goal of this paper is to quantify that performance and to propose enhancing mechanisms. We do not attempt a user study of whether the performance is adequate.

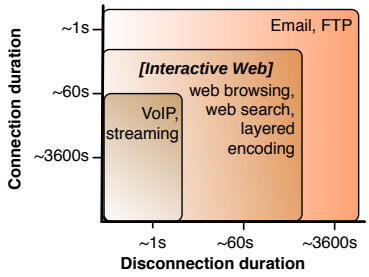
Cellular networks today follow a different business model. A massive tower infrastructure investment is recovered through new user subscriptions. In areas without enough users willing to pay the monthly fees, the infrastructure investment is not profitable. Rural and developing areas that want Internet access may find WiFi infrastructure quicker and cheaper to deploy even if performance or convenience is initially poorer compared to cellular. Thus, providing cellular access is a top-down decision, whereas WiFi enables a cheaper and more organic model of evolution. Moreover, for large organizations over moderate areas (e.g, municipalities and campuses that cover 2–10 square miles), a mesh- or AP-based WiFi network is more economically feasible. The cost of WiFi scales better per user than the linear costs of purchasing cellular access for every users. For example, our small municipality cannot afford 3G subscriptions for every worker as it is an expense that carries over to each year's budget; however, a one-time purchase of WiFi equipment is feasible. In fact, in Amherst, MA, the primary argument for deploying a mesh network was to avoid wired phone lines on Town equipment that cost $40/month each for every budget year.

Mobility supported by WiFi presents many challenges. At one end of the spectrum are densely deployed networks, where in core areas, nodes have full connectivity and high link quality. However, even in such well-provisioned environments, supporting demanding applications is difficult. For example, VoIP can perform poorly because of the requirements for low packet loss, delay, and jitter [5]. At the other end, in areas where infrastructure is unavailable, only DTN-based options are feasible and they work only for non-interactive applications [4]. This paper addresses a middle ground, where APs are available but insufficient for supporting the most demanding applications, but may attempt to provide more than email or bulk transfer. Such scenarios can occur in organically deployed networks or at the fringes and occluded portions of dense WiFi networks. Below, we outline this network design space in more detail and state our design goals.

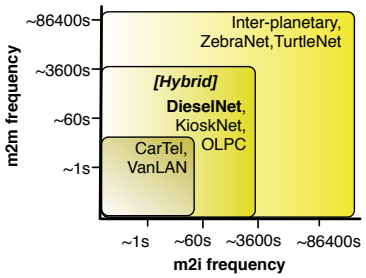
2.1 Design space

Figure 1(a) illustrates the applications targeted by Thedu, namely, the middle region marked *Interactive Web*, primarily referring to Web search and browsing. When disconnection periods are less than a second, it is possible to support applications like VoIP [5]. When disconnection durations are large or unpredictable, the range of applications that can be supported today is limited to those that only need a file transfer abstraction, e.g., email, limited use of FTP or HTTP, or sensor data collection [16].

Web search and browsing are inherently interactive in nature, but do not have the stringent delay requirements of applications such as VoIP. However, these web applications cannot be supported today, when connectivity is intermittent. Consider a tech-savy taxi passenger, Alice, who is visiting Springfield and wishes to find a sushi bar. Alice does not have a cellular data plan and instead takes advantage of the city's free WiFi. When within range of an AP, she sends the keywords "springfield sushi sake" to her favorite search engine. Before she can receive a response, the AP is out of range causing TCP to stall. She refreshes her browser to send the query again and receives a sorted list of Web links and snippets. She retrieves the top URL, but unfortunately it's for a restaurant in a Springfield in the wrong state, and while she is reading, the taxi has driven away from the AP. She waits for the next AP to retrieve the second URL or perhaps modify her query. Thedu seeks to improve user-perceived per-



(a) Application space in terms of connection duration.



(b) The mobile design space in terms of m2i and m2m meeting frequency.

Figure 1: Design space: Interactive web applications in a Hybrid Infrastructure-DTN

formance or robustness for interactive applications for exasperated vehicular Web users like Alice.

Figure 1(b) characterizes different vehicular environment in terms of m2m and m2i meeting frequency. Thedu focuses on the *hybrid infrastructure-DTN environment* in the middle. When APs are densely deployed, mobile nodes encounter an AP frequently [16, 28, 25], and m2m meetings provide little additional benefit. However, when the deployment is sparse, exploiting m2m benefits can provide improvement in capacity or user-perceived application performance. KioskNet [34], One Laptop Per Child [1], and Saami [33] target such environments. Such efforts do not have the resources to deploy cellular towers, but may be able to deploy cheap WiFi APs.

2.2 Goals

Thedu's architecture is guided by the following design goals.

- *Robustness*: Vehicular WiFi connectivity is prone to disruptions, so applications designed for persistent connectivity must be adapted to degrade gracefully.
- *Low-cost deployability*: The system should be low-cost and need minimal change to existing infrastructure or software.
- *Extending access*: Many rural or developing regions have limited access points to the Internet, so the system should extend the Internet's reach.

The Thedu design uses two key ideas: *(i)* aggressive prefetching to transform Web search and browsing to a one-shot request response format; *(ii)* leveraging m2m contacts when appropriate to further improve user-perceived application performance. In the next section, we describe the Thedu system, that enables Web search from vehicles. In Section 4, we describe an extension to Thedu, that exploits m2m contacts for performance benefits.

3. WEB SEARCH WITH Thedu

Thedu uses a proxy to aggressively prefetch web pages from a search engine and transforms the Web search process to a one-shot

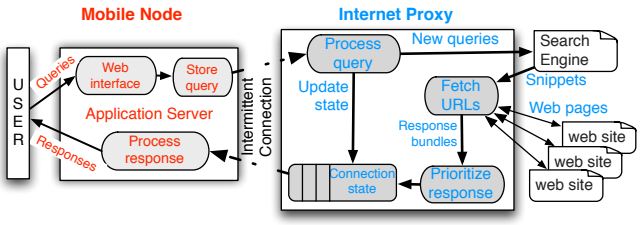


Figure 2: Thedu system architecture

request response format. Figure 2 illustrates the Thedu system. A server-side proxy awaits a connection from a mobile. If the mobile has pending web pages, it downloads them immediately upon connection. In parallel, the proxy interacts with the Web search engine on behalf of the mobile to retrieve web pages for new queries. A similar proxy-based architecture has been used by several prior systems [16, 12] to make (non-interactive) applications more robust to disconnections.

The mobile accepts search queries from a user through a Web interface. The queries are sent to the proxy with a nonce that is used to re-identify the query after a disconnection. The proxy interfaces with a search engine, such as Google, Yahoo or MSN, and retrieves URLs and document snippets sorted by their relevance to the query. The proxy *prefetches* the web page bodies associated with the URLs in parallel. Studies have observed that 75% of users do not view more than the top 20 web pages and most users are interested in at most 5 results per query [18], so our implementation only prefetches the top 20 web pages. When the proxy receives multiple queries from a mobile, it retrieves web pages for each query in parallel. The proxy also retrieves all images and other objects and places them in a per-user queue. The mobile user then downloads the web pages from the queue.

Our IR contribution is a novel set of techniques to process prefetched web pages, so that a mobile user downloads a significantly greater number of relevant web pages and avoids wasting bandwidth on web pages that are unlikely to be useful.

3.1 IR meets networking

Aggressive prefetching makes the demand for bandwidth unlimited, so Thedu uses a prioritization protocol to allocate limited bandwidth to the most useful web pages. First, using a simple classifier, Thedu identifies queries that likely require only a single web response. For these queries, the Thedu proxy only returns the top few web pages. For web pages of the remaining queries, Thedu computes the relevance probability; i.e., the probability that the web page is relevant to the query. Thedu prioritizes the web pages based on their relevance probability.

3.1.1 Query-type classification

Understanding the user's intent by classifying the query type can help increase the usefulness of web pages and limit useless prefetching. For example, returning only one web page suffices for a user's query of "Mobicom 2008", even if other web pages have a high relevance score. Broder [8] provides a nomenclature for classifying search queries as *homepage*, *content* and *service* queries. *Homepage* queries try to find known items; e.g., "cnn". *Content* queries seek an answer or a meaning and not a specific website; e.g., "kosovo conflict". *Service* queries are informational.

Thedu classifies queries as either homepage or non-homepage using a simple Bayesian classifier. We enumerate characteristic features of homepage and non-homepage queries and use these fea-

Homepage	Content
Query terms/acronyms occur in URL	Query is a question
All query terms occur in title and anchor text	One of the top-3 URLs is a *wiki*
Query is less than 3 words	Query is greater than 3 words
URL is a root	

Table 1: Features to classify query type

tures to train the classifier. The features are summarized in Table 1. Note that the URL, title, and snippet fields are short and easy to parse. So, Thedu can classify query-type in real time. When tested on Google with 50 training queries and 100 test queries, Thedu is able to predict homepage queries with 90% accuracy and non-homepage queries with 71% accuracy. We describe more details about the query-type classifier in our earlier work [6]. From an IR standpoint, our contribution is novel as the Thedu classifier works in real time without knowledge of the entire web collection, unlike existing query-type classifiers [21].

3.1.2 Prioritizing web pages

Thedu prefetches several web pages for queries, but the client only has limited bandwidth to download them. Today's search engines do not prioritize web pages across different queries as they are typically not as constrained by bandwidth as intermittently connected networks. For example, is the fifth web response to query A with a relevance score of 4.3 more or less important than the seventh response to query B with a relevance score of 8.2? Thedu nodes must answer this question in order to prioritize web pages across queries. We develop information retrieval (IR) techniques to compute normalized relevance scores that are comparable across queries.

Thedu's normalization algorithm is based on the Kullback-Liebler divergence. We include details of the normalization technique in the Appendix. Thedu then computes relevance probability given the relevance score, and prioritizes web pages according to the relevance probability.

The relevance probability is computed as follows. Let $P(score|rel)$ denote the probability of score given the document is relevant and $P(score|nonrel)$ the probability of score given the document is not relevant. Manmatha et al. [26] analyze the score distribution and suggest that scores of a relevant document, $P(score|rel)$, follows a gaussian distribution and scores for non-relevant documents, $P(score|nonrel)$, follow an exponential distribution. We estimate the parameters for the two distributions using a training data set of 70 queries and retrieved web pages that were already classified as relevant and non-relevant. We then estimate the relevance probability using Bayes' rule.

$$P(rel|score) = \frac{P(score|rel)P(rel)}{P(score|rel)P(rel) + P(score|nonrel)P(nonrel)} \quad (1)$$

For a detailed description of the relevance probability computation, see our earlier work [6].

3.2 Scope and limitations

Several applications other than Web search can benefit from a design similar to Thedu to adapt to intermittently connected networks. For example, several commercial accelerators for Web browsing prefetch hyperlinks on a page to reduce user-perceived response time. Popular prediction algorithms use a Markov model [30] to estimate the probability that a user clicks on a hyperlink, which naturally serves as a utility in Thedu. Li et al. [23] present a utility-driven atmospheric sensing system that uses progressive compression to send the most important data first under limited bandwidth, which is crucial for hazardous weather prediction applications like tornado detection. Similarly, layered-encoded multimedia can be streamed to vehicular users for buffered playback by prioritizing base layers.

PRIORITIZE (r_i, r_j):

1. *If r_i and r_j are both not destined to X*:
   (a) Set $U(r_i) = P(r_i) \cdot Q_X(r_i)$ and $U(r_j) = P(r_j) \cdot Q_X(r_j)$
   (b) Return web page with higher utility.
2. *If r_i is destined to X and r_j is not*:
   (a) If number of replications of $r_i = 0$, return r_i.
   (b) Else, set $U(r_i) = P(r_i) \cdot (1 - Q(r_i))$, $U(r_j) = P(r_j) \cdot Q_X(r_j)$.
   (c) Return web page with higher utility.
3. *If both are destined to X*: Send web page with higher relevance probability

Figure 3: Thedu prioritization at the proxy

The Thedu design cannot improve performance for interactive applications that do not permit prefetching. Our implementation of Thedu does not support dynamic content, mobile code, or interactive Ajax applications. Such applications require a more sophisticated proxy, engineering which is outside the scope of this paper.

4. M2M ROUTING

Thedu leverages m2m contact to route web pages to other mobile nodes. As a first step, during a m2i contact with the proxy, mobile nodes download web pages that need to be routed to others, apart from downloading their own web pages. The natural question is, If a mobile client has pending web pages, should it still download web responses for other peers? This notion is counter intuitive. Typically, in the DTN [4, 9, 35, 17] and MANET [31, 20] literature, the primary focus is to deliver a packet to its destination at the first available opportunity. However, for Web search, we find that downloading peer responses, even when there are pending web responses for one-self, can provide substantial benefit. The reason is that the relevance probability is a concave function, i.e., responses have decreasing marginal utility and most of the utility resides in the first few web pages.

4.1 m2m routing at proxy

The goal of Thedu's routing protocol is to maximize the number of relevant web pages delivered by the deadline. Let $P(r_i)$ be probability that web page r_i is relevant. Let $Q(r_i)$ be the probability that the web page r_i will be delivered within the deadline by the AP, and $Q_X(r_i)$ be the probability that the web page r_i will be delivered by node X within the deadline. Web pages are routed using prioritization as follows: when X meets an AP, the proxy ranks two web pages r_i and r_j as shown in Figure 3. PRIORITIZE returns r_i if r_i has higher priority than r_j and vice versa.

In Step 1 of PRIORITIZE, Thedu prioritizes responses for peer nodes according to the utility U. The utility r_i is defined as the probability that the web page is relevant and will be delivered within the deadline, i.e., the product of the relevance probability and the

delivery probability. Between delivering a pending web page r_i and routing another peer's web page r_j (Step 2 in PRIORITIZE), Thedu does the following: if the pending web page r_i has never been routed though any peer node, then the client downloads the web page. If r_i has already been routed through a peer but is not yet delivered, Thedu estimates the utility of downloading r_i, which is the probability that the web response will miss its deadline. The utility of the other web page r_j is same as the first case. If both web pages are pending for X (Step 4 in PRIORITIZE), Thedu prioritizes the web pages according to the relevance probability.

4.2 m2m routing at client

Routing between two clients is also based on prioritization. Suppose mobile node Y has an m2m contact with mobile node X. Node Y prioritizes all web pages to be delivered to X first, using the relevance probability. Y prioritizes the remaining web pages using the utility function

$$U(r_i) = P_r(r_i) \cdot Q_X(r_i) \tag{2}$$

Eq. 2 represents the probability that the web page r_i is relevant and will be delivered within the deadline by node X.

Both the prioritization protocol at the proxy as well as at the client use the probability that a web page is delivered within a deadline. To estimate this probability, the Thedu node estimates the expected meeting time with all other nodes as an average of past meetings. Some nodes may never meet; in this case the expected meeting time is set to *infinity*. Thedu makes a simplifying assumption that the meeting times are exponentially distributed with the parameter equal to the expected meeting time. Accordingly, Thedu computes the probability that the web page is delivered within the deadline as $1-e^{-\frac{1}{\lambda}d}$ where λ is the expected meeting time and d is the deadline. We use an exponential distribution because vehicular meeting times in our testbed are difficult to model [38]. Approximating inter-meeting time to an exponential distribution simplifies delay computation and works well in practice [4].

4.3 Exploiting query locality

Thedu uses caching to further benefit from m2m contacts. Web queries are known to exhibit locality [37] and the query frequency follows Zipf's law. Thedu nodes exploit this domain knowledge by caching all web pages they route. On meeting a peer node, the node returns web pages from its cache if there is a hit. The proxy tracks the popularity of queries based on its frequency and marks each response web page with an indicator of the popularity of the query. This indicator allows the clients to remove web pages for unpopular queries from the cache, if necessary.

4.4 Modeling m2m benefit

In this section, we model the benefits of using m2m contact opportunities. Our model is specifically designed for cases when the utility of downloading a response is defined by a concave function. When the utility is linear, the m2m benefit is simply the additional throughput available from m2m contacts. In recent related work, Banerjee et al. [7] quantify the delay benefits of m2m routing for linear utilities.

Concave utilities provide a unique advantage for m2m routing. Since downloading subsequent web pages have decreasing utility value, the overall utility of the system increases when nodes route web pages for others, rather than only download their own web pages. Our model quantifies the benefit of m2m routing in terms of m2m versus m2i meeting frequency. Finally, we use our model to show that, for an example concave utility function, m2m routing is only beneficial when the mobile nodes meet other peers with the same frequency as they meet APs.

n	Number of nodes
a_i	Inter-meeting time between node i and AP
b_i	m2m inter-meeting time between i and a peer
x_i	Number of responses transfered by i in a m2i meeting
y_i	Number of responses transfered by i in a m2m meeting
$U_i(x)$	Utility to node i on receiving x responses
H	Utility with only m2i contacts
G	Utility when m2m contact is leveraged

Table 2: Model parameters.

Table 2 lists our notation. We make the following assumptions. Each node's demand is infinite, i.e., a node can use all available AP bandwidth to fetch responses of decreasing utility. All responses are of unit size. The experiment runs for unit time.

If nodes only use contact opportunities with APs, then the total utility, H, is

$$H = \sum_{0<i\leq n} U(\frac{x_i}{a_i}) \tag{3}$$

as each node i on average meets an AP $1/a_i$ times and fetches x_i responses in each meeting.

Let G denote the total utility when nodes use m2m contact opportunities in addition to m2i contact. Assume that responses routed through a node i for node j will be delivered if sufficient m2m contact bandwidth is available. Let e_i denote the total number of responses received by i. If $U_i(\cdot)$ is assumed to be concave and smooth, the total utility G is maximized when e_i is chosen such that the derivative $U_i'(e_i)$s are equal. For simplicity, we assume that all $U_i(\cdot)$ are uniform and are denoted by $U(\cdot)$. G is estimated as follows.

$$G = max \sum_{i\in n} U(e_i) \tag{4}$$

$$\sum_{i\in n} e_i \leq \sum_{i\in n} \frac{x_i}{a_i} \tag{5}$$

$$e_i \leq \frac{y_i}{b_i} + \frac{x_i}{a_i} \tag{6}$$

Eq. 5 states that the total number of responses received by all nodes is at most the total AP bandwidth. Eq. 6 says that node i receives at most as many responses as allowed by its AP opportunity plus its *useful* m2m opportunity $\bar{b}_i$. That is, when i meets another node carrying responses destined to i. Finally, just the throughput benefit T of m2m routing is

$$T = \frac{\sum_{i\in n}(y_i/b_i)}{\sum_{i\in n}(y_i/b_i + x_i/a_i)} \tag{7}$$

LEMMA 1. *For uniform utilities, m2m routing provides benefit, i.e., $G > H$, only if not all $x_i, 1 \leq i \leq n$ are equal. That is, some nodes have lower m2i contact opportunity than others.*

Assuming the above condition is satisfied, the following theorem computes G. Without loss of generality, assume that nodes $1 \ldots n$ are sorted in increasing order of $\frac{y_i}{b_i} + \frac{x_i}{a_i}$. Let f_i denote $\frac{y_i}{b_i} + \frac{x_i}{a_i}$.

THEOREM 1. *G is maximum for the smallest k such that $e_i = f_i$, and $e_{k+1} = \ldots = e_n = \frac{1}{n-k}(\sum \frac{x_i}{a_i} - \sum_{1\leq i\leq k} e_i)$ satisfies Eq. 6, and such a k is well-defined.*

The above theorem gives a simple procedure for computing G, namely going from $k = 0$ to n or until the condition in the theorem is satisfied. The theorem also suggests that m2m routing can yield

benefit due to two fundamental reasons: *(i)* concave utilities and *(ii)* workload skew.

Concave utilities The decreasing marginal utility of responses implies that it is more beneficial for some nodes with high x_i to fetch responses for other nodes than to download their own responses. If m2m contacts were over provisioned, then G is maximum when all e_i's are identical.

When m2m contacts are constrained, we compute $\overline{b}_i$ and G as follows. Per Lemma 1 above, suppose m nodes satisfy $x_i \leq \frac{1}{n}\sum \frac{x_i}{a_i}$ and the remaining $n-m$ nodes have higher capacity. A m2m meeting is useful only if one of the former m nodes meets one of the latter $n-m$ nodes. If m2m inter-meeting times are uniform, $\overline{b}_i$ is $\frac{(n-m)b_i}{n}$. Theorem 1 then gives $e_i, 1 \leq i \leq n$ that maximizes G.

Workload skew The model so far assumed that all nodes have infinite demand with decreasing marginal utilities. If we relax the assumption and assume that each node i demands at most d_i responses. Assume that d_i is less than x_i for some nodes, and greater or equal for the rest. Then, the total utility using only APs is $H = \sum_{0<i\leq n} U(min(d_i, \frac{x_i}{a_i}))$.

Without loss of generality, assume nodes $1 \ldots n$ are sorted in increasing order of $f_i = min(d_i, \frac{b_i}{\overline{b}_i} + \frac{x_i}{a_i})$. Suppose m nodes have d_i greater than x_i. As before, if node i meets all other nodes with a uniform inter-meeting time $\frac{b_i}{n}$, then $\overline{b}_i = \frac{(n-m)\cdot b_i}{n}$. Theorem 1 then gives the set of e_i's that maximize G. Note that skew in the workload alone is sufficient for m2m routing to be beneficial even if the utilities are concave. A concave utility function implies that the allocation that maximizes G is unique.

4.4.1 Numerical example

We solve for H and G for a sample network configuration. This example lets us study the relationship between the m2m routing benefit and the ratio of m2i and m2m contact opportunity.

As before, we assume that some nodes, that we call the fringe nodes, have smaller m2i contact opportunity than others. Let nodes 1 to m be the fringe nodes with no m2i contacts and let the ratio of m2i and m2m contact frequency be α. We study the effect of α and m on m2m benefits.

Let the m2i contact frequency be a. Then the m2m contact frequency is $\alpha \cdot a$, for $\alpha > 1$. Let the capacity of nodes $m+1 \cdots n$ be x; i.e., nodes can transfer x responses during an m2i contact. The capacity of all m2m contacts is also x. For this numerical example, we define the utility $U(x) = \sum_{i=1}^{i} \frac{1}{x}$.

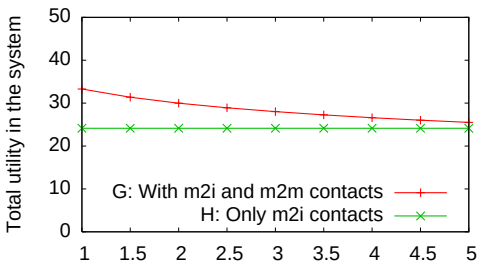
Using Eqs. 3–6 and Theorem 1, we have the following.

$$H \approx (n-m) \cdot \log(\frac{x}{a})$$

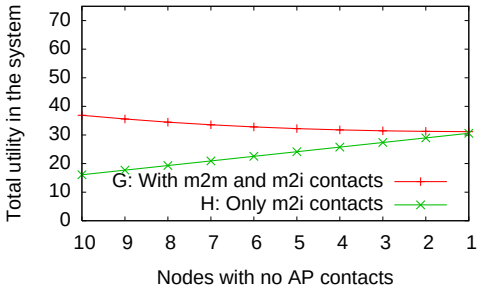
$$G \approx m \log(\frac{nx}{(n-m)\alpha a}) + (n-m) \cdot \log(\frac{x}{a} - \frac{nc}{\alpha a (n-m)^2})$$

We set $n = 20$, $x = 1$ and $a = 0.2$. Figure 4(a) shows the utility of H and G for varying α for our chosen utility function. When m2m contacts are as frequent as m2i contacts, the benefit of m2m routing is 40%. However, the benefit reduces quickly to less than 5% when m2m contact frequency is 5 times smaller than m2i contact frequency. Figure 4(b) shows the utility H and G as the number of fringe nodes m decreases. Under a uniform mobility model and uniform AP placement, m decreases as APs are added to the infrastructure, so Figure 4(b) also says that as more APs are added to the infrastructure, m2m benefit reduces from over 100% to less than 5%.

5. DEPLOYMENT



(a) *Increasing* α, $m = 10$



(b) *Decreasing m,* $\alpha = 1$

Figure 4: Quantifying m2m benefits for an example network.

We deployed Thedu and compared it to a simple stateless proxy on our hybrid Infrastructure-DTN testbed in Amherst, MA. Our DTN testbed, DieselNet [3] is comprised of 40 buses, of which an average of 21 buses are on the road. Each bus is equipped with a linux computer and an 802.11b radio that constantly scans for open APs and other buses. For a more detailed characterization of the testbed, see Burgess et al. [9].

In our deployment, the buses carry an implementation of the client-side Thedu and connect to the server-side proxy Thedu, as shown in Figure 2. We also deployed a *stateless* proxy in our testbed as a point of comparison.

We conducted experiments using the deployed Thedu and the *stateless* proxy, each for 5 days from March to April 2007. The two proxy experiments differ as follows.

- The *Thedu proxy* (without m2m routing) is implemented as described in Section 3. The web pages are prioritized using relevance probability.
- The *stateless proxy* strips features from the Thedu proxy so that it is equivalent to the case where mobile nodes do not use a proxy. The stateless proxy retrieves web pages for queries and returns bundles to the client in FIFO order (i.e., by completed retrieval time). It terminates all incomplete transactions when the client disconnects; retrievals begins anew upon reconnection.

5.1 Experimental setup

Recall, from Section 3 that the Thedu proxy interfaces with an already available Web search engines (e.g., Yahoo, Google, or MSN) to retrieve the URLs of top web pages, and then it prefetches the contents of the web pages. For our deployment we chose to use Indri [36], an academic Web search engine. An important difference between commercial search engines and Indri is that the latter indexes a large collection of static web pages; commercial search engines, on the other hand, actively crawl the web. Two features

Collection:	W10g [2]
Number of TREC queries:	150
Queries:	TREC 2001
Search engine:	Indri
Query deadline:	30 min
Queries per hour:	10 per bus

Table 3: IR parameters used for deployment.

of Indri make it more appropriate for evaluating Thedu than a commercial search engine.

1. *Indri allows evaluation of retrieval performance:*The IR community and NIST's TREC [2] have built a standard web collection. The web collection has predefined user queries and human *relevance judgments* for each query; i.e., a list of web pages in the collection that are relevant to the query. Indri indexes this static web collection. The relevance judgment allows us to evaluate the performance of Thedu in terms of the number of relevant web pages delivered to the user. Note that the judgments are used *only* for evaluation.

2. *Indri provides response scores:* Indri assigns a relevance score for the web pages. Thedu uses the relevance score to estimate the relevance probability, which is subsequently used to prioritize web pages. Commercial search engines use (non-normalized) relevance scores, but only return a ranked list of web pages.

The reason that we do not provide an interface from Thedu to a commercial API (e.g., Google) is that we would not be able to present evaluation results without a large study of user experiences. In contrast, by using Indri and TREC we control repeatable experiments that leverage past user studies of search relevance. In future work, we plan to deploy Thedu with the Google API using a simple heuristic to translate the rank to a relevance probability.

We used Indri to index and store a standard collection of web pages from the TREC WT10G web collection [2]. We used standard queries from 2001 TREC web-track associated with the WT10G collection, for evaluation. We evaluate the retrieval performance using the relevance judgments. This query set and evaluation technique is commonly used in the IR community to measure retrieval effectiveness. We modified the Indri source code to provide normalized scores as described in the Appendix. For both proxies, we removed queries and associated web pages after 30 min (since we assume most passengers would exit the bus by that point). The deployment parameters are tabulated in Table 3.

The queries associated with the W10G collection are limited and relevance judgments are only available for the TREC prescribed queries. The vehicular client periodically generates a *(queryID, query)* pair, where queryID is a monotonically increases sequence number and query is chosen from the predefined query set. The queryIDs allow a query to be repeated as if it were completely new, and allow us to evaluate performance for a large number of queries.

For both deployment experiments, queries are generated with an average of 10 per hour per bus with inter-arrival times drawn from an exponential distribution. At each AP opportunity, the bus connects to the proxy and sends all queries generated since the last connection. The buses periodically upload statistics of queries, web pages, and delays. Delays are calculated against each query's generation time stamp, so they include the time waiting for the next AP.

5.2 Deployment results

Statistics	Thedu Proxy	Stateless Proxy
Number of meetings	897	935
Total queries:	780	743
Total web pages returned:	5639	1207
Avg resps. per query:	7.2	1.6
Relevant web pages:	1630	401
Queries with at least 1 relevant web page:	529 (68%)	291 (39%)

Table 4: Deployment: Average per day statistics during the two different weeks of deployment

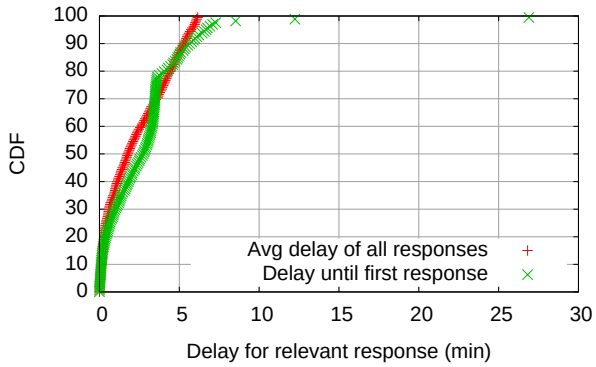


Figure 5: Deployment: CDF of the average delay in receiving relevant web pages.

The results of our deployments are presented in Table 4. Each result is a per-day average. Thedu was able to return more than 4.5 times as many web pages on average. More importantly, by prioritizing web pages according to the relevance probability, the number of relevant web pages sent by Thedu is 4 times larger than the stateless proxy. Finally, Thedu was able to send at least one relevant web page for twice as many queries compared to the stateless proxy. We measure the delay of receiving a relevant web page for a query. Figure 5 shows an empirical CDF of the delay of receiving relevant web pages at the client. We note that 90% of the time, the first relevant web page is received within 5 min. The mean delay is receiving the first relevant web page is 2.7 min and the mean delay in receiving all relevant web pages is 2.3 min. In Section 6.2, we show that the average delay in receiving relevant web pages is a function of AP density.

6. TRACE-DRIVEN SIMULATIONS

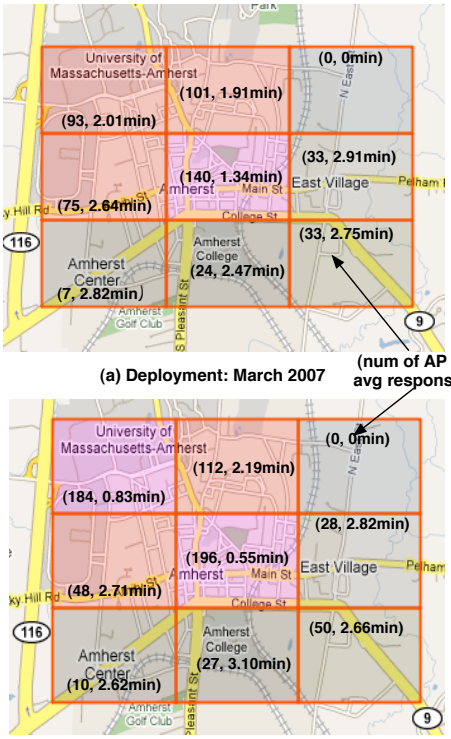
We collected traces of vehicular connectivity in DieselNet to perform trace-driven studies that augment our deployment results. In particular, we use the traces to

- study the trends in AP density and its effect on Thedu performance,
- evaluate the benefits of m2m routing,
- and evaluate the performance of Thedu for Web browsing, instead of Web search.

We first present our trace collection methodology.

6.1 Trace collection and experimental set up

Each bus in the testbed constantly scans for open APs and other buses. If an AP is found, our software caches the DHCP lease to



(a) Deployment: March 2007

(b) Trace: November 2007

Figure 6: Deployment and Trace: Number of m2i contacts and average delay classified according to geographical locations. The map is a 2 sq.mile snapshot of the DieselNet testbed around Amherst Town center.

speed up the IP address acquisition process in the future. To determine if the connection is usable, the bus pings a known Internet server.

If another bus is found, the connection lasts until the radios are out of range. During contacts, buses transfer random data, to measure the capacity of the connection. However, buses do not transfer data during m2i contacts to avoid overloading open APs operated by third parties; instead the buses frequently ping a known server to estimate contact duration. The buses log information about both m2i and m2m contacts. The buses log the identity of the remote connection (SSID), the location and duration of contact, and for a m2m meeting, the number of bytes transfered. The logs are periodically uploaded to a central server. All trace-driven simulation results in this paper are based on traces collected between October 22 to November 18, 2007, which resulted in 20 days of trace data by excluding weekends and holidays.

The experimental set up for the trace-driven simulations is similar to the deployment set up, described in Section 5.1. Each node simulates query generation and the inter-query time is drawn from an exponential distribution. The proxy retrieves the top 20 webpages from the Indri search engine. We compute the number of relevant responses delivered using the relevance judgement. The contact time and transfer size between two buses is given by the trace data. In the case of AP contacts, the trace provides contact duration but not the transfer size, because we do not send data during m2i contacts. We set the bandwidth for m2i contacts to 205 KBps, which is the bandwidth of m2m contacts (see Table 5). The deadline for each query is set to 30 min.

Statistic	m2m	AP
Avg unique nodes:	21 (buses)	151 (APs)
Number of contacts:	242	4964
Avg contact duration (sec):	10.3	15.17
Avg bandwidth (KBps):	204.9	N/A
Total transfer (MB):	482	15, 071 (at 205 KBps)

Table 5: The characteristics of m2i and m2m contacts.

6.2 Effects of AP density on Thedu

Intuitively, in Thedu, the delay in receiving a web response is dependent on the density of APs. We evaluate this hypothesis using two experiments. First, we divide the deployment region into grids that have different AP density and study Thedu performance in each of the grids. The grid is centered at the Amherst town center and spans about 1 mile on each side. Though the DieselNet deployment spans a larger area, we concentrate on a snapshot of the deployment region that has a high frequency of buses.

Second, we compare the deployment results with trace-driven simulations from *recent* traces. In August 2007, a town mesh network was added to the DieselNet testbed. Therefore, in the recent trace data, AP connectivity in the town center has increased, compared to during the deployment.

Figure 6 is a heat map of m2i contact frequency of the Thedu deployment in March 2007 and using 5 days of traces collected in November 2007. Buses meet APs more frequently in November for two grids compared to during the deployment: in the town (center grid) and near the university (top left grid). AP meeting frequency in the other grids does not change significantly between the deployment and trace. The meeting frequencies refer to per-day frequencies.

Next, we compute the average delays for responses delivered in each grid. In Figure 6, the average delay to deliver relevant responses is greater than 2.53 min in grids where the number of m2i contacts is less than 60. The delay reduces to less than 1.51 min when the number of m2i contacts becomes greater than 120. In the November traces, Thedu delivers relevant responses with a delay of 0.55 min in the Amherst town center. There are about 25 unique APs in the town center excluding the mesh.

Though our measurement study shows that AP density has been increasing since March 2007, the trend may not be universal. In DieselNet, m2i meetings have increased partially due to the deployment of a mesh network. However, our study shows that as the number of AP meetings increase, the average response delay decreases. In a separate trace-driven experiment (not shown here), we find that the delay in receiving the first relevant response in the town center is 1.67 min. During the deployment, this delay was 2.11 min.

6.3 Evaluation of m2m benefits

In Section 4, we described an extension to Thedu to exploit m2m contacts using routing. In this section, we evaluate the benefits of m2m routing for our vehicular testbed using trace-driven simulations.

6.3.1 Contact characteristics

Table 5 tabulates per-day statistics collected from the trace data. Access points are an inexpensive commodity product, and securing their access is harder than allowing public access. Hence, there were 151 open APs in our environment put up by third parties. Network-equipped buses are an expensive research platform, and the buses in our testbed were deployed by the authors only. Accordingly, our testbed has an imbalance. Given the APs found on

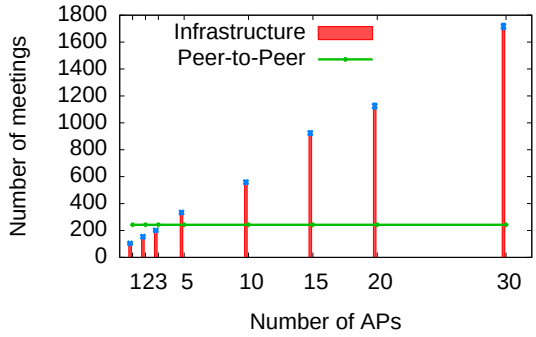


Figure 7: The m2i and m2m contact counts for varying number of APs.

the field, the ratio of m2m contacts to m2i contacts is 1:20. In terms of throughput, m2m routing can at most provide a benefit of $\frac{482}{15,071} = 3.2\%$ (refer to "total transfer" in Table 5). It is clear that even in a semi-urban area such as Amherst, m2m routing can provide little benefit.

Next, we evaluate the benefits of m2m routing when only a small number of APs are available. For example, in OLPC-like environments, AP deployment is sparse, but mobile laptop devices can help deliver web pages to clients that are multiple hops from the APs. Figure 7 shows the ratio of m2m and m2i contacts for varying number of available APs over a single day. In this experiment, APs were chosen randomly from the 151 APs. The m2m-to-m2i contact ratio is 1:1.3 when only 5 APs are available for contact, compared to 1:20 when 151 APs are available.

6.3.2 *Benefits of m2m routing*

To evaluate the benefit of m2m routing, we compare three Thedu variations: *(i)* no m2m routing *(ii)* m2m routing *(iii)* caching and m2m routing. In our caching experiments, we set the query repeat rate conservatively to 20%. That is, 20% of queries generated by each bus is picked from a pool of 50 queries. The remaining 80% are unique. In the web, the observed query repeat rate is much higher: 30 to 40% [37]. In all our experiments, we randomly choose 5 APs from the available 151 APs. The results are averaged over 10 seeds and error-bars indicate 95% confidence interval.

Figure 8 shows the number of relevant web pages delivered for all buses for varying load. In this setting, m2m routing provides up to 28% benefit over not using m2m routing. When web pages are cached, up to 58% more relevant web pages are delivered compared to the base case.

M2M routing is most useful for buses that meet the APs infrequently. Figure 9 shows the performance of the three Thedu variations for these *fringe* buses: 30% of the buses with the smallest number of m2i contacts. As expected, m2m routing delivers on an average 48% more relevant web pages within a deadline for fringe buses compared to the base case. The caching benefit is even higher: up to 90%. For high loads, the performance of all three systems saturate because the contact opportunities have limited size.

Figure 10 shows the CDF of the delay in receiving relevant web pages. The mean delay is 6.7 minutes, three times higher compared to when all APs were available for contact. In rural scenarios, even though m2m routing delivers more relevant web pages, the delays may deter users from using WiFi for Web search. An increase in m2m contacts or higher AP density can reduce this delay and make Web search deployment using open APs practical for rural areas.

6.4 Web browsing with Thedu

Like Web search, Web browsing is an iterative, interactive process; however, browsing begins with a URL and the response is a single web page. To make this process transactional, the Thedu proxy prefetches hyperlinks and returns the corresponding web pages to the mobile client in one shot. If the user subsequently requests the prefetched hyperlink, then interactivity is reduced markedly.

We evaluate the performance of Web browsing using traces from our testbed. Specifically, we determine the average time to receive a response. Since *click through* data about which links a user follows for different web pages is not publicly available from search engines, we use the TREC collection of 50 queries as a replacement. We treat each query as a URL and the 10 top-ranked web page responses for each as the hyperlinks for that URL.

The probability of a user requesting a hyperlink is known to follow Zipf's law [11] and hyperlink prefetching algorithms exhibit more than 80% accuracy in prediction [11, 30]; thus, we assume that the probability of requesting a hyperlink follows the Zipf's law. We have 50 URLs for the simulated browsing application, which we artificially increase by using a <URL, ID> pair as the identifier, and we create new IDs as needed.

We simulated a version of Thedu for Web browsing. Each bus generates web requests at a rate of 5 requests per 5 minutes. On receiving the response page, the bus requests up to 3 hyperlinks per page. The buses choose the hyperlinks according to Zipf's law; i.e. the most popular hyperlink has a higher probability of being chosen. At the proxy, in response to a web request, Thedu returns the requested page as well as the prefetched hyperlinks. The prefetched hyperlinks are prioritized according to the probability that the hyperlink will be requested. The deadline for each request is 30 minutes. Using our 20 days of trace data, we found that if the proxy prefetched responses, the average delay in receiving a response is 1.1 min. On the other hand, if we used a *stateless* proxy as described in Section 5, the average delay is 3.9 min. Thedu can provide 4 times speed up compared to a *stateless* proxy, for Web browsing.

7. RELATED WORK

Thedu enables popular *interactive* Web applications to tolerate disruptions in connectivity and uses m2m DTN routing to augment sparse AP deployment. To this end, we draw upon ideas from a large body of prior work as described below.

System architecture There appears to be a consensus on using a proxy-based architecture to tolerate disruptions in vehicular connectivity. Seth et al. [34] present an architecture and prototype for using vehicular mobility and stationary kiosks to extend the reach of sparsely deployed Internet gateways to a much larger rural area. They use an Internet proxy to hide disconnection from legacy servers and pick a nearby proxy to improve TCP throughput. Ott et al. [29] use a proxy to hide disconnections and bundle a web page and inline objects into a single file to improve performance similar to HTTP 1.1 with pipelining. Hull et al. [16] present a mobile sensor computing system where vehicles sense and submit data to a central database that serves as a Web portal. Eriksson et al. [12] use a proxy to improve TCP throughput by distinguishing between losses in the wired and wireless halves of a connection. In comparison to these works, we build a proxy to aggressively prefetch web pages and enable search for mobile nodes.

Measurement Bychkovsky et al. [16] study the achievable throughput and meeting frequency of open WiFi APs in Boston from vehicles. Zhang et al. [38] study inter-vehicular connectivity in the DieselNet [3] testbed and conclude that it is not always predictable. Others have conducted detailed studies of TCP perfor-

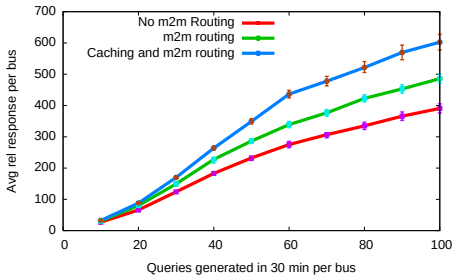


Figure 8: 5 APs: Exploiting m2m contacts provides up to 26% benefit averaged over all buses. Caching provides up to 58% benefit over the base case.

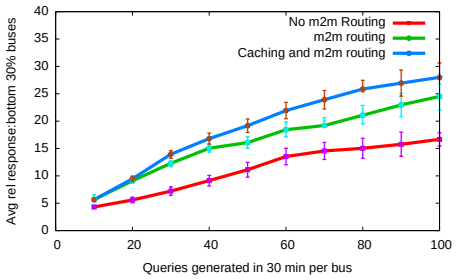


Figure 9: 5 APs: m2m routing provides an average of 48% improvement in relevant documents for *fringe* buses that meet APs infrequently.

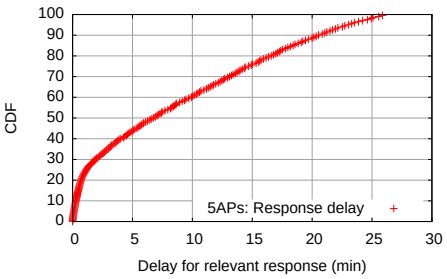


Figure 10: 5 APs: CDF of the delay is receiving a relevant response for all queries. The mean delay is 6.5 minutes.

mance [15] or link quality characteristics [25] under smaller controlled vehicular testbeds. Our experiments in Section 6.3 are based on open APs that were deployed organically; our focus is on the architecture and routing algorithms to enhance interactive applications, not on transport or link quality measurement.

Prefetching Prefetching to mask disconnection was explored in the context of hoarding and trickle re-integration in the Coda file system [22]. Chandra et al. [10] used prefetching to improve availability or response time for disconnected Web operation. Padmanabhan and Mogul [30] proposed a markov model for predicting Web requests. The TEK [32] system is an email-based web browser that compresses prefetched search results into an email. Jiang and Kleinrock [19] proposed a technique where a connected client prefetches files during idle time, based on the probability that the user will request the file. In comparison, Thedu prefetches all of the top web pages, but uses IR techniques to prioritize, so that clients download the most relevant web pages during short periods of connectivity.

Intermittently connected networks Thedu is similar in spirit to the *infostation* [14] paradigm where base stations store information for mobile nodes, and a mobile node downloads the information when connected. Goodman et al. [14] quantify the benefit of this architecture and its incentive scheme based on an analytic model and synthetic traces for a file distribution application. PeopleNet [27] presents a query/response (precise) matching service in a virtual social network of mobile people. Thedu has a complementary focus on interactive Web applications in a hybrid infrastructure/DTN environment.

In our previous work[6], we demonstrated that Web search can be enabled from moving buses using opportunistic WiFi APs. In comparison, we move beyond infrastructure APs and quantify the added benefit of m2m routing and study the effect of AP density on Web search. DTN routing has seen a flurry of research activity in recent times (see [17, 35, 4, 24]). Much of this work considers an ad hoc collection of nodes meeting intermittently. Such scenarios are useful for military, first-response, underwater sensors, animal monitoring, and other infrastructure-less environments.

8. CONCLUSIONS

The ubiquity and low cost of WiFi is an attractive alternative to high bandwidth cellular data services such as 3G, which is a popular wireless choice today. However, it is unclear if interactive web applications can be supported using intermittent WiFi connectivity. Our research goal is to understand how to architect interactive applications to be robust to disconnections. Our position is that building and deploying real applications is a valuable first step towards this goal.

We developed Thedu, a system to enable interactive web search application in intermittently connected environments. Thedu uses aggressive prefetching to mask disconnection. We deployed Thedu on DieselNet, our vehicular testbed in Amherst, MA. Our results indicate that Thedu can deliver on average four times as many responses compared to not using Thedu. We then asked how applications can be architected to gracefully degrade from infrequent disconnections to long periods of disconnections with an AP. In particular, we analyzed if and when mobile-to-mobile routing can yield nontrivial benefit in application performance compared to only using mobile-to-AP contact opportunities. Our analytic model and trace-driven simulations suggest that routing yields little benefit even under moderately dense AP deployment such as in Amherst. With sparsely deployed APs, mobile-to-mobile routing with caching improves the number of relevant responses delivered by up to 58%, but the delays are significantly higher, limiting its applicability to delay-tolerant applications even in rural settings.

Acknowledgments We thank Bruce Croft and Yun Zhou for collaborating on an early version of this work [6]. We also thank Brian Lynn for his work on the DieselNet system.

9. REFERENCES

[1] One Laptop Per Child. `http://laptop.org/`.
[2] Text Retrieval Conference (TREC). `http://trec.nist.gov`.
[3] UMass DieselNet. `http://prisms.cs.umass.edu/dome`.
[4] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM Sigcomm*, pages 373–384, August 2007.
[5] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. Levine, and J. Zahorjan. Interactive WiFi Connectivity For Moving Vehicles. In *Proc. ACM Sigcomm*, August 2008.
[6] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani. Web Search From a Bus. In *Proc. ACM Workshop on Challenged Networks (CHANTS)*, pages 59–66, September 2007.
[7] N. Banerjee, M. Corner, D. Towsley, and B. N. Levine. Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proc. ACM Mobicom*, September, 2008.
[8] A. Broder. A Taxonomy of Web Search. *SIGIR Forum*, 36(2):3–10, 2002.
[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE Infocom*, April 2006.
[10] B. Chandra, M. Dahlin, L. Gao, A.-A. Khoja, A. Nayate, A. Razzaq, and A. Sewani. Resource Management for Scalable Disconnected Access to Web Services. In *Proc. Intl World Wide Web Conf.*, pages 245–256, May 2001.
[11] X. Chen and X. Zhang. A Popularity-Based Prediction Model for Web Prefetching. *Computer*, 36(3):63–70, 2003.

[12] J. Eriksson, S. Madden, and H. Balakrishnan. Cabernet: A Content-Delivery Network for Moving Vehicles. In *Proc. ACM Mobicom*, September 2008.
[13] Fon. `http://www.fon.com/`.
[14] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates. Infostations: A New System for Data and Messaging Siervices. In *Proc. Vehicular Technology Conference*, pages 969–973, May 1997.
[15] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *Proc. ACM Mobisys*, pages 206–219, June 2007.
[16] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: a Distributed Mobile Sensor Computing System. In *Proc. ACM SenSys*, pages 125–138, October 2006.
[17] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. ACM Sigcomm*, pages 145–158, 2004.
[18] B. Jansen, A. Spink, and T. Saracevic. Real Life, Real Users, and Real Needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.
[19] Z. Jiang and L. Kleinrock. Web prefetching in a mobile environment. In *IEEE Personal Communications*, volume 5, pages 25–34, September, 1998.
[20] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
[21] I.-H. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In *Proc. ACM SIGIR*, pages 64–71, July 2003.
[22] J. J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. In *Proc of the thirteenth ACM symposium on Operating systems principles*, pages 213–225, 1991.
[23] M. Li, T. Yan, D. Ganesan, E. Lyons, P. Shenoy, A. Venkataramani, and M. Zink. Multi-user Data Sharing in Radar Sensor Networks. In *Proc. ACM SenSys*, pages 247–260, November 2007.
[24] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. In *Proc. SAPIR Wrkshp*, 2004.
[25] R. Mahajan, J. Zahorjan, and B. Zill. Understanding WiFi-based Connectivity From Moving Vehicles. In *Proc. ACM IMC*, pages 321–326, October 2007.
[26] R. Manmatha, T. Rath, and F. Feng. Modeling Score Distributions for Combining the Outputs of Search Engines. In *Proc. ACM SIGIR*, pages 267–275, September 2001.
[27] M. Motani, V. Srinivasan, and P. Nuggehalli. PeopleNet: Engineering a Wireless Virtual Social Network. In *Proc. ACM Mobicom*, pages 243–257, August 2005.
[28] J. Ott and D. Kutscher. A Disconnection-Tolerant Transport for Drive-thru Internet Environments. In *Proc. IEEE Infocom*, March 2005.
[29] J. Ott and D. Kutscher. Bundling the Web: HTTP over DTN. In *Proc. Workshop on Networking in Public Transport*, August 2006.
[30] V. Padmanabhan and J. Mogul. Using Predictive Prefetching to Improve World Wide Web Latency. In *Proc. ACM Sigcomm*, pages 22–36, July 1996.
[31] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.
[32] W. T. S. A. Libby Levison. Searching the World Wide Web in Low-Connectivity Communities. In *2002 International Symposium on Technology and Society*, June 2002.
[33] The Sami Network Connectivity Project. `http://www.snc.sapmi.net`.
[34] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *Proc. ACM Mobicom*, pages 334–345, September 2006.
[35] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. ACM Workshop on Delay-Tolerant Networking*, pages 252–259, August 2005.
[36] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A Language Model-Based Search Engine for Complex Queries. In *Proc. Intl. Conf. on Intelligence Analysis*, May 2005.
[37] Y. Xie and D. R. O'Hallaron. Locality in Search Engine Queries and Its Implications for Caching. In *Proc. IEEE Infocom*, pages 1238–1247, June 2002.
[38] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. ACM Mobicom*, pages 195—206, September 2007.
[39] Y. Zhou, B. N. Levine, and W. B. Croft. Distributed Information Retrieval For Disruption-Tolerant Mobile Networks. CIIR Technical Report IR-412, University of Massachusetts Amherst, 2005.

10. APPENDIX

To normalize response scores across all queries, we use the negative Kullback-Leibler divergence between the query and the response. This removes dependance of the specific query in the score estimation. We explain the query normalization technique as a modification of the Indri [36] search engine. Indri [36] is an academic search engine shown to be effective for web search. Indri returns a ranked set of documents in response to a query and associates each document with a relevance score. In case of web search, a document is a web page.

In the Indri model, the *relevance score* of a document D for query Q is estimated as

$$Score(D) = \prod_{w \in Q} \lambda P(w|D) + (1-\lambda)P(w|C) \tag{8}$$

where λ is a smoothing constant, $P(w|D)$ is the probability of a word w occurring in a document D and $P(w|C)$ is the probability of the word w occurring in C, the entire collection of documents being searched.

Since the document score depends on the words in the query and the collection, the scores of documents for different queries (or over different collections) are not comparable. To normalize document scores across queries we propose a score normalization method that computes scores as:

$$Score(D) = \frac{1}{|Q|} \sum_{w \in Q} log \frac{\lambda P(w|D) + (1-\lambda)P(w|C)}{P(w|C)} \tag{9}$$

Note that the document rank remains unchanged by normalization, as does the search effectiveness. In our deployment, we set $\lambda = 0.4$. A full derivation and explanation of Eq. 9 is available in a technical report [39].